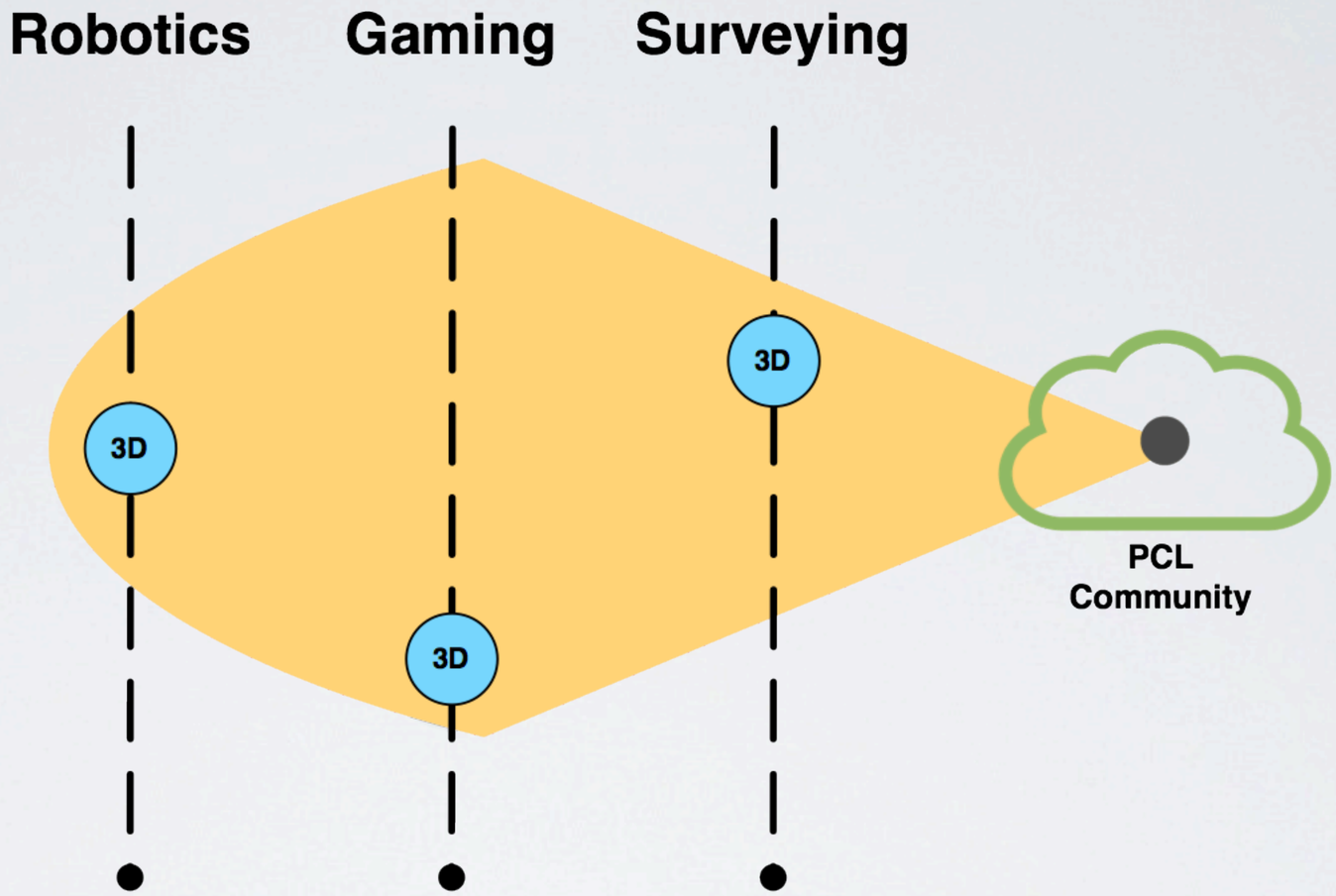
POINT CLOUD LIBRARY

Radu B. Rusu
Alexandru E. Ichim

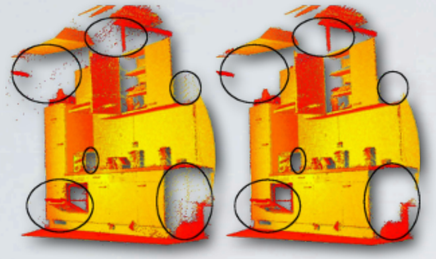
Product development as it is:

Development as it should be:

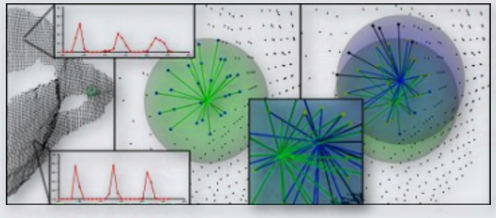
Motivation



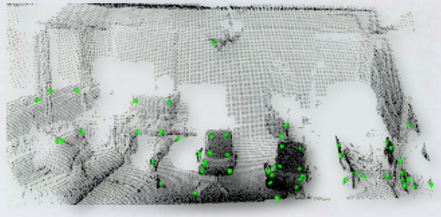
Motivation



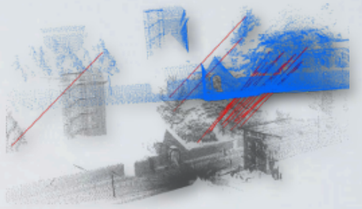
Features



Filters

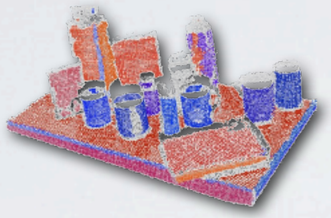


Keypoints

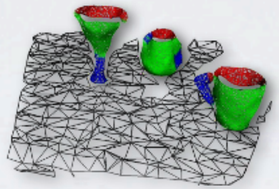


Registration

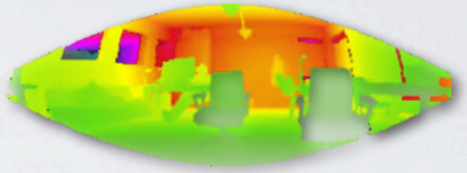
Segmentation



Sample Consensus

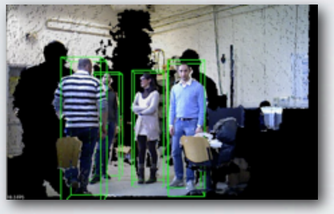


Surface

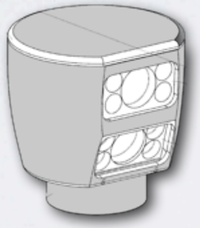


Range Image

I/O



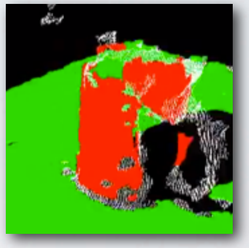
People



Simulation

Out-of-core

Visualization

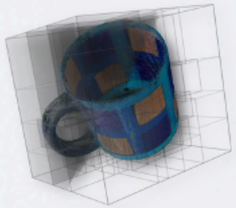


Segmentation

2D

ML

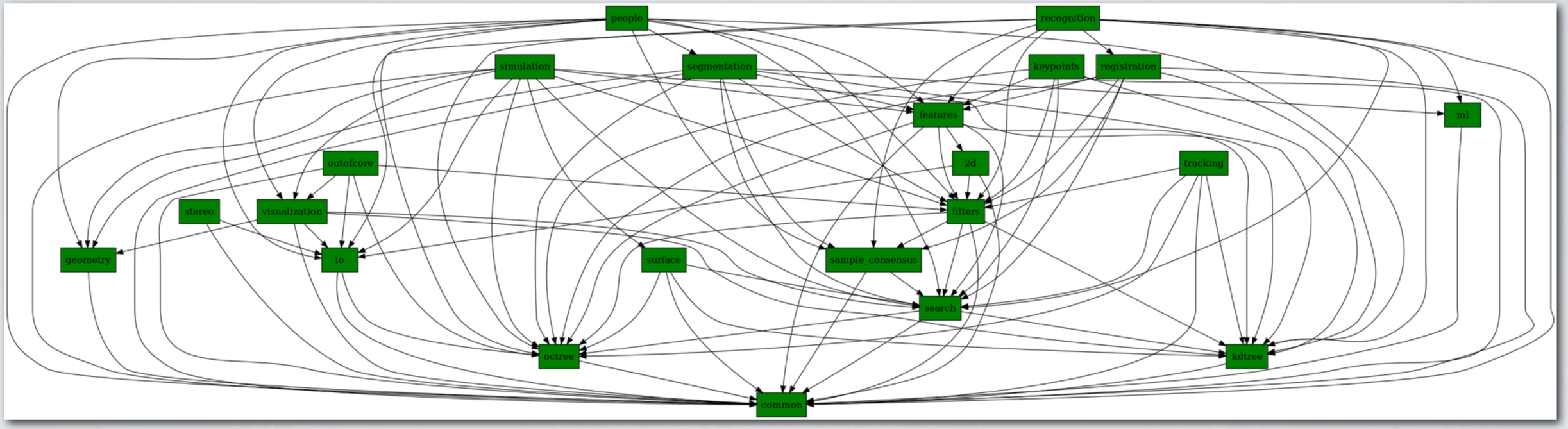
PCL Modules



Kdtree

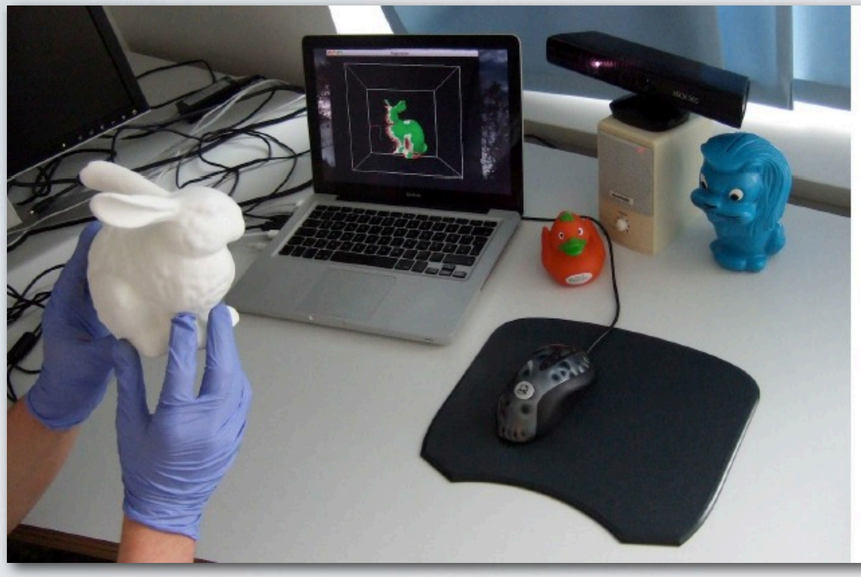
Octree

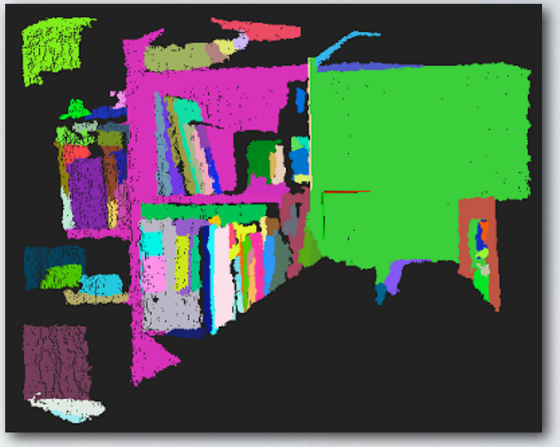
Module dependencies

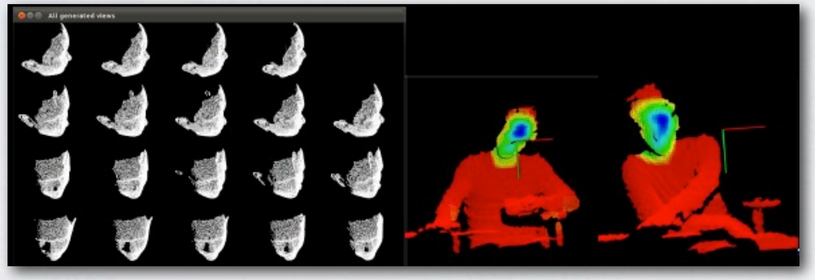


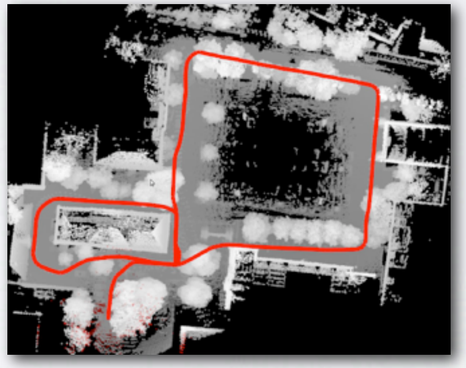
PCL Modules

And a lot of **demo apps** to show off the technologies!

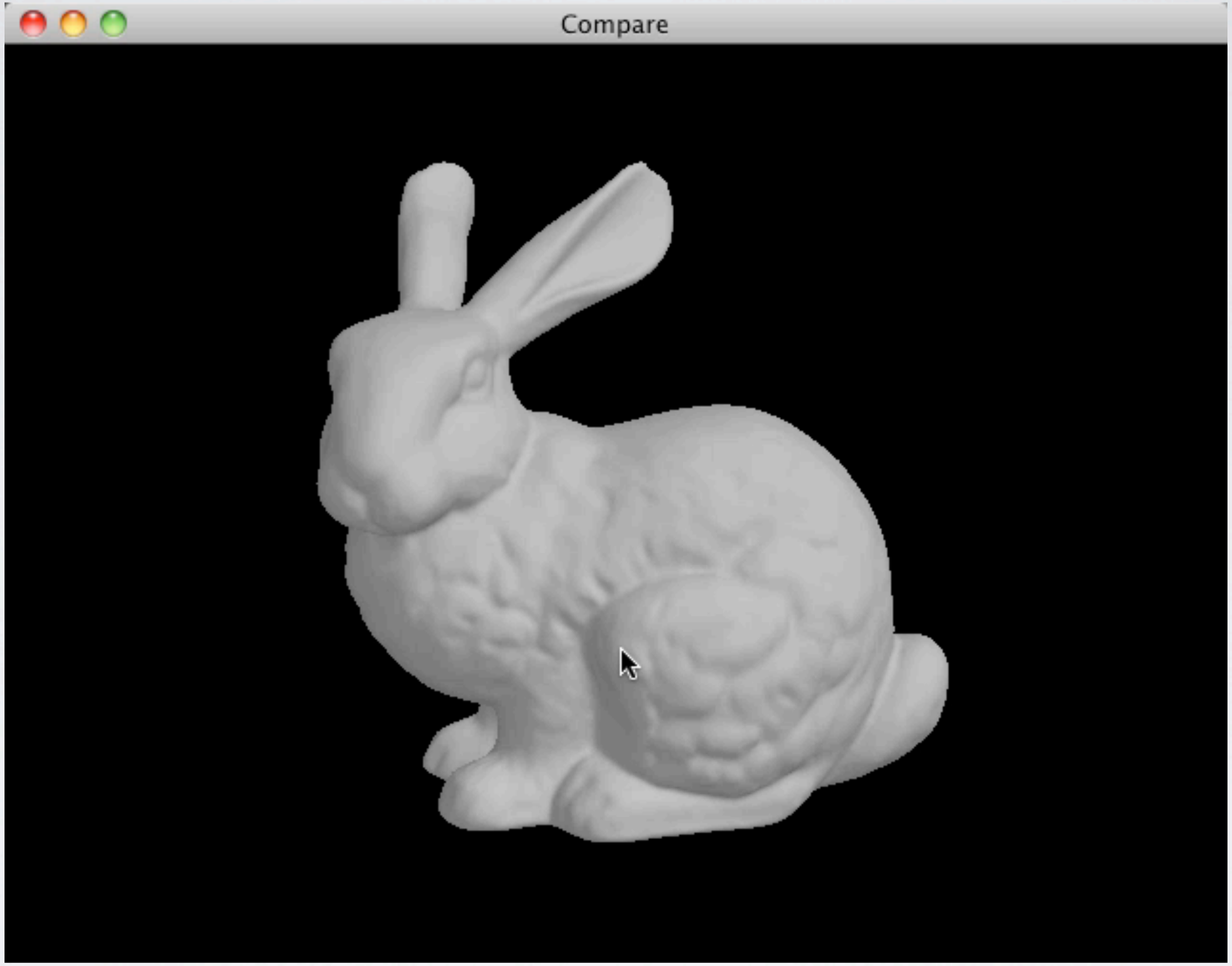








PCL Modules


Compare

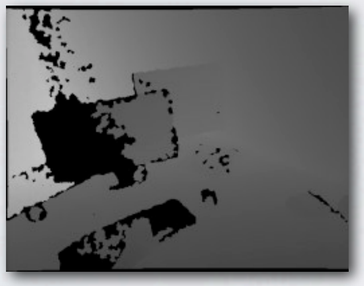

(1.11402)

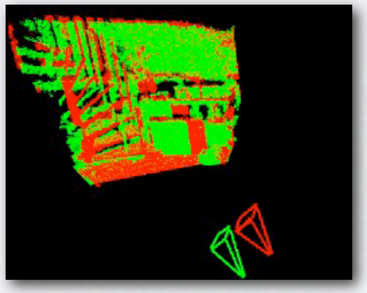
Capture
Bilateral filtering
Normals

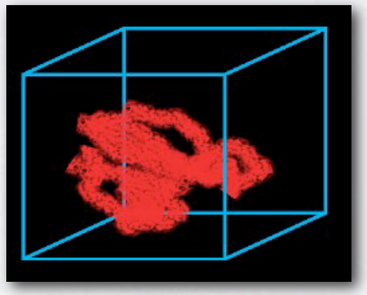
Frame-to-frame
camera tracking

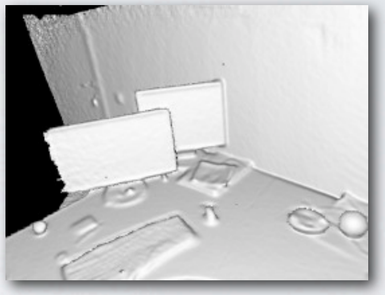
Volume
Integration

Ray casting



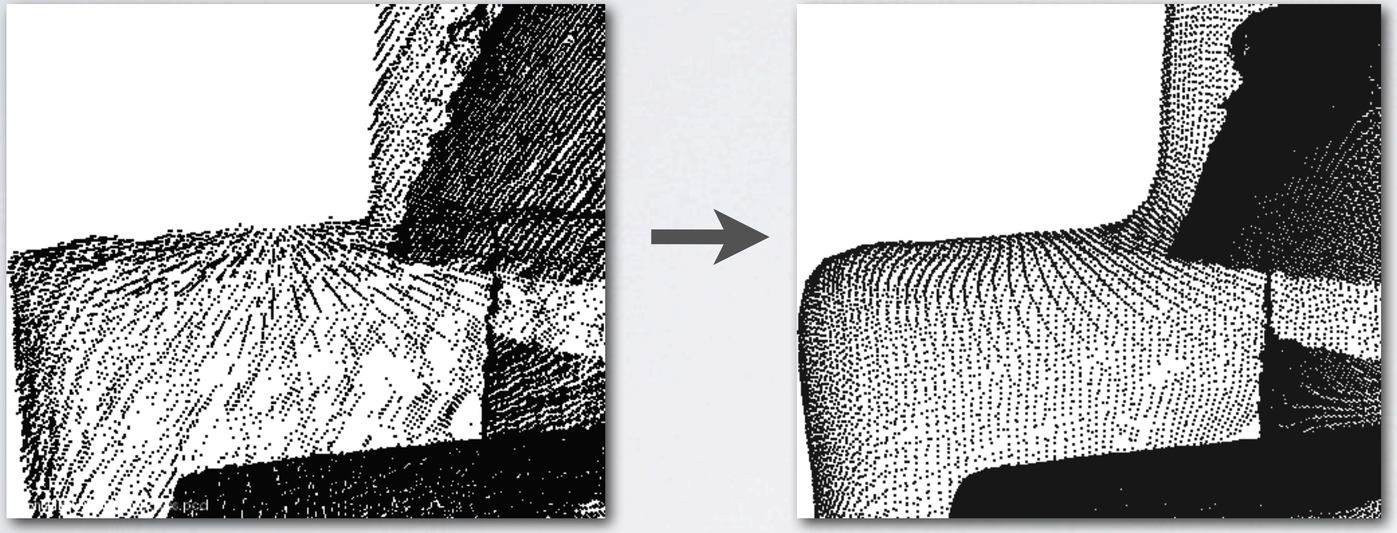




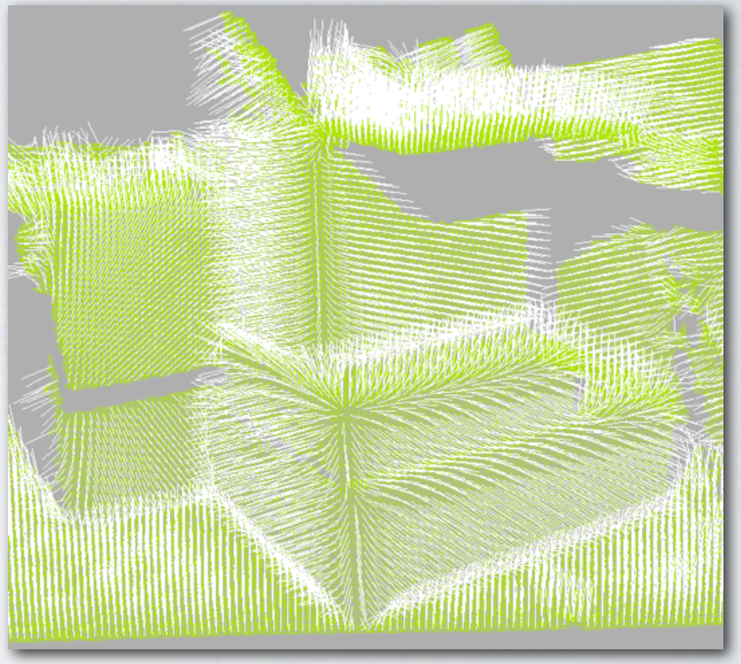


KinFu 1/4

Capture
Bilateral filtering
Normals



noise removal via bilateral filtering

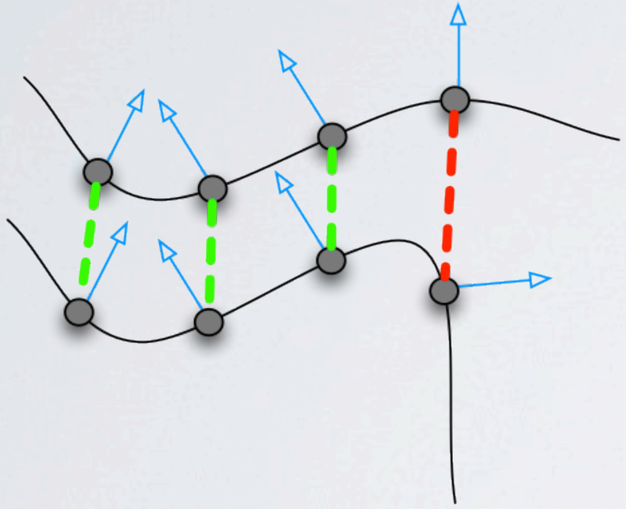


highly-parallelizable normal estimation

RGB-D data is an "organized" raster

Frame-to-frame camera tracking

Want to compute relative transformation between consecutive frames



Projective point correspondences using normal compatibility

$$\sum_{i=1}^{N} ((R\,p_i + t) - q_i) \cdot n_{q_i})^2$$

Point-to-plane error metric

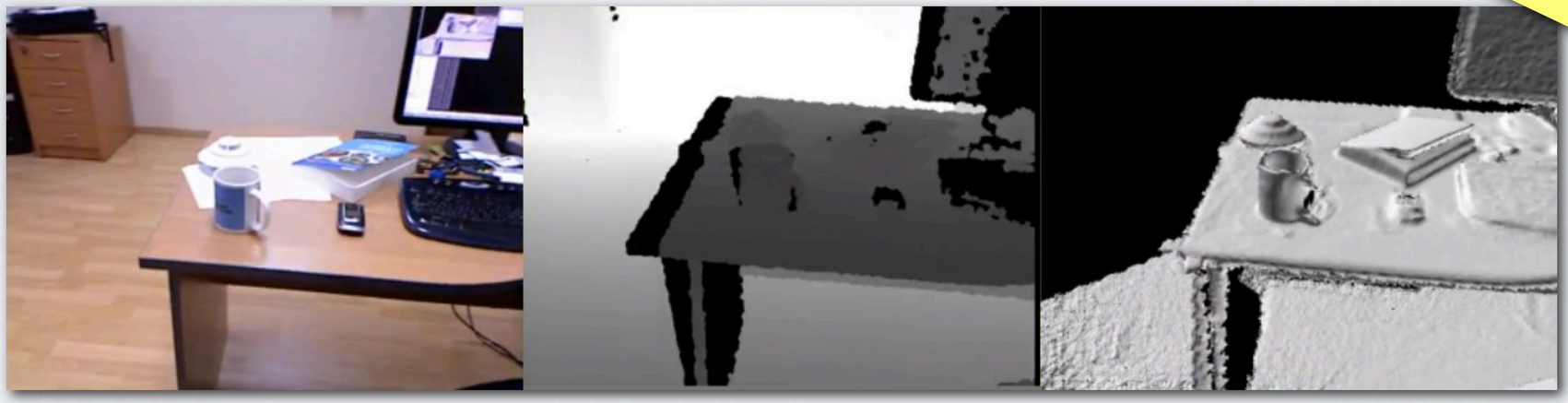
Over-determined system

Solve it in the least squares sense

Volume Integration

Incrementally store the incoming frames

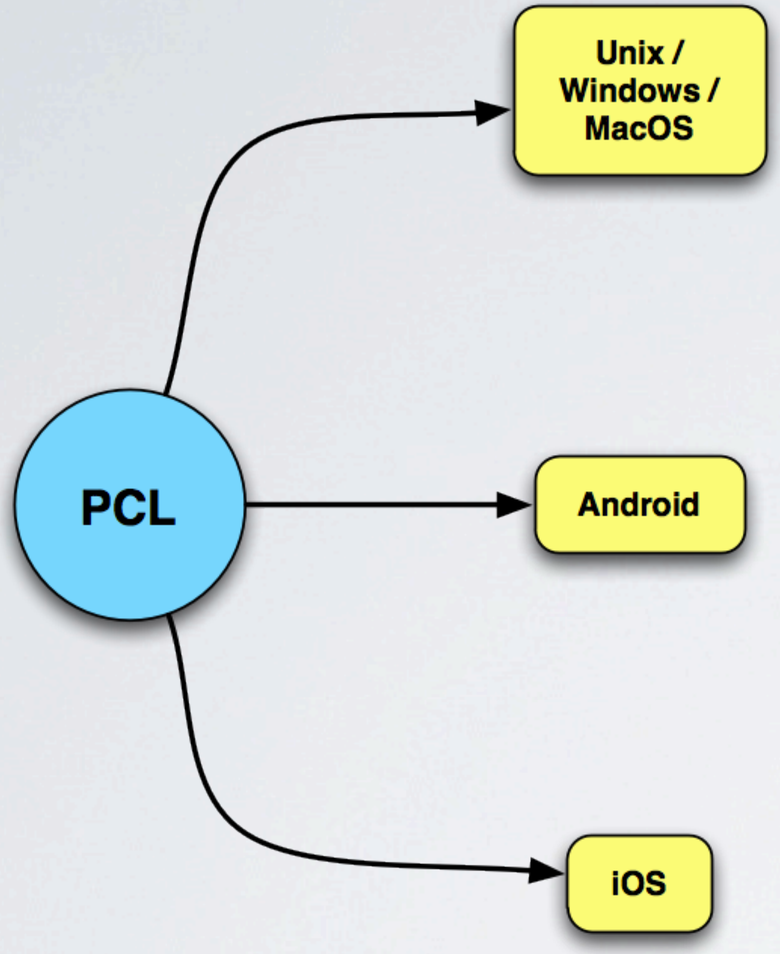
Truncated Signed Distance Function (TSDF)



for each voxel in the TSDF
    project it to the RGB-D frame
    update depth from pixel - running average

Ray casting

KinFu 4/4



Everything works !

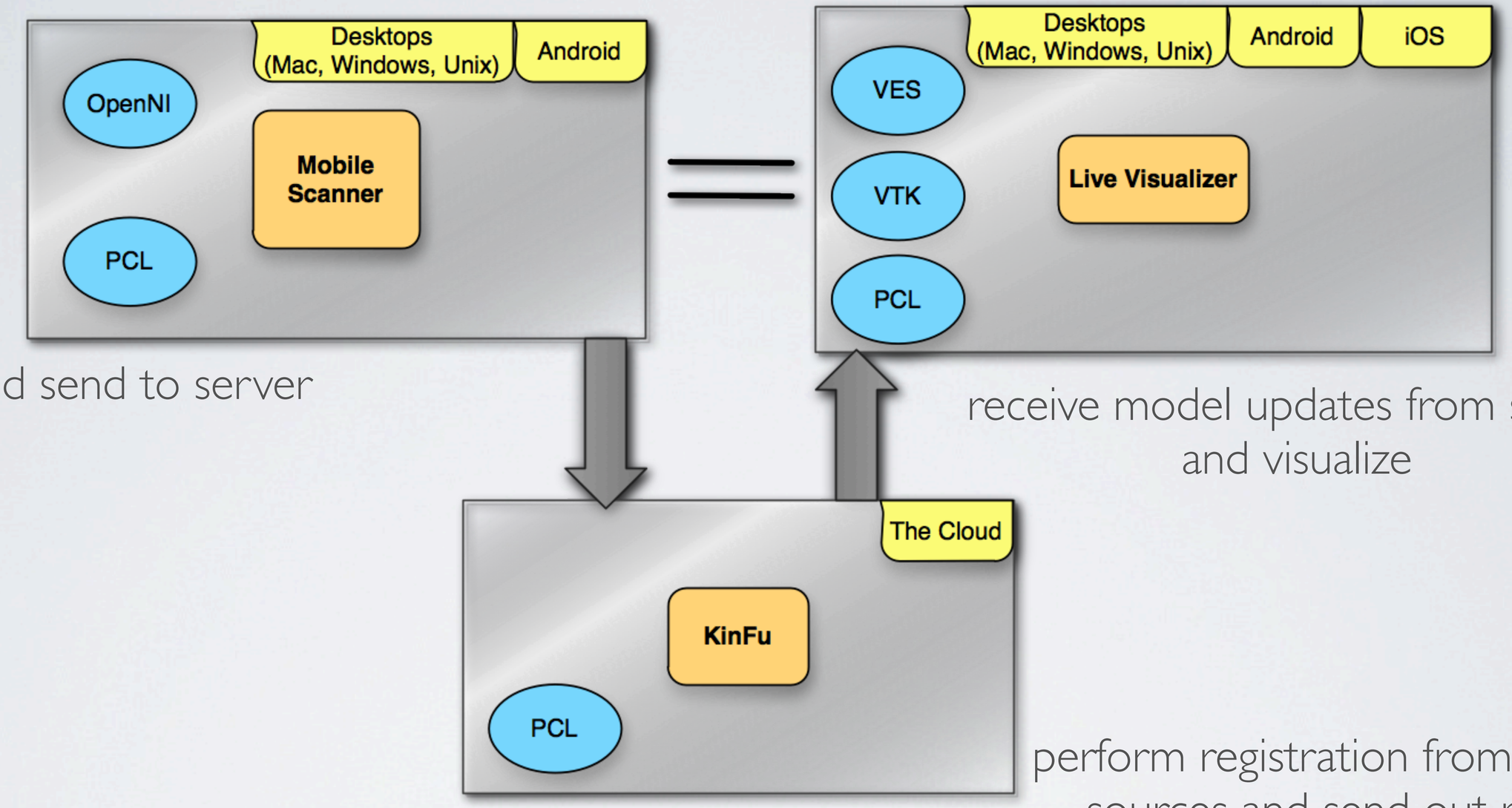
- support for multiple sensors
- processing
- visualization
- CUDA implementations

- visualization with OpenGL / VTK / OpenSceneGraph / ...

- visualization with OpenGL ES and VES

- Apple does not allow unlicensed 3rd party accessories
- visualization with OpenGL ES and VES

PCL platforms

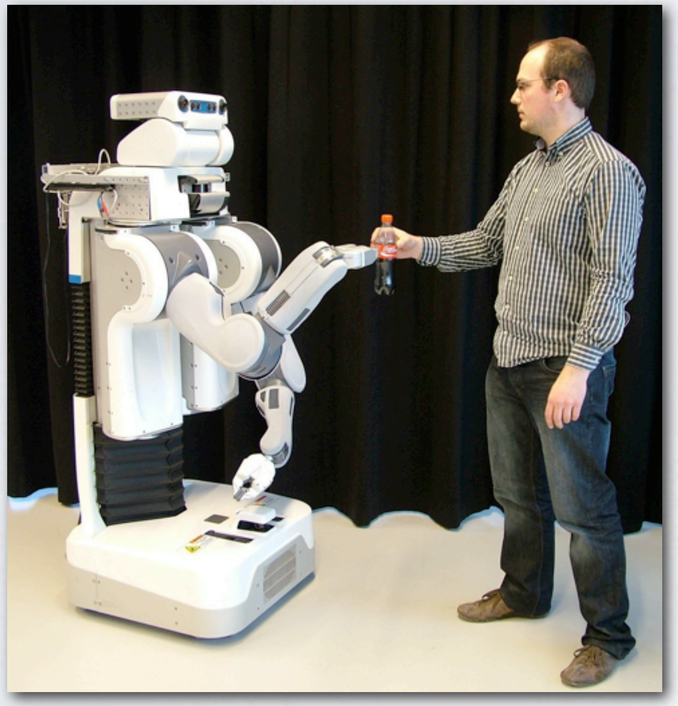


Mobile Mapping in the Cloud

DEMO TIME !

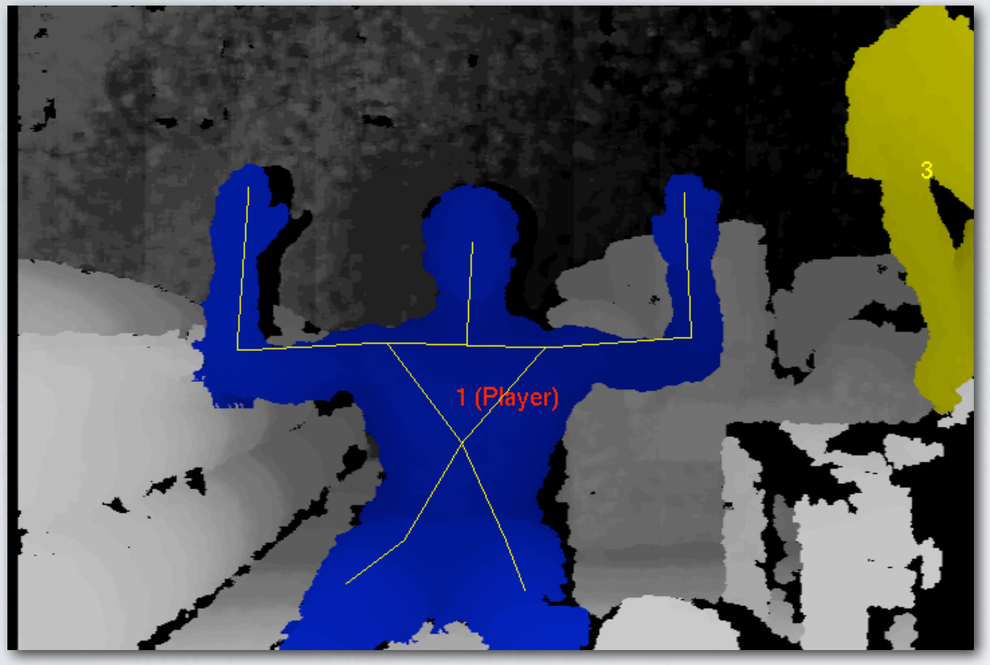
WHY?

- easier and more precise in 3D
- both pose detection and tracking



A lot of applications:

- human-robot interaction
- medical applications
- entertainment
- education
- ...



1. OpenNI (NITE)

- needs background subtraction (i.e., segmentation)
- based on a skeleton model
- assumes fixed camera

People Tracking

Other Systems 1/2

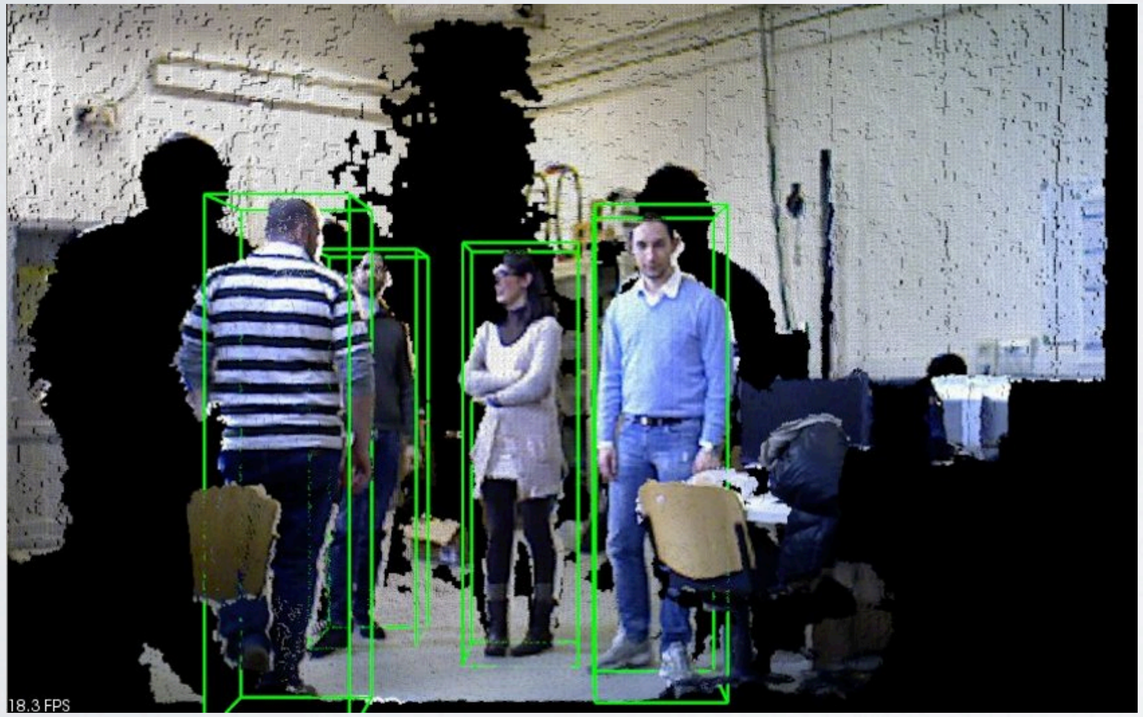
2. XBox

- needs background subtraction (i.e., segmentation)
- does not use a model
- assumes fixed camera
- classifies each pixel

People Tracking

Other Systems 2/2

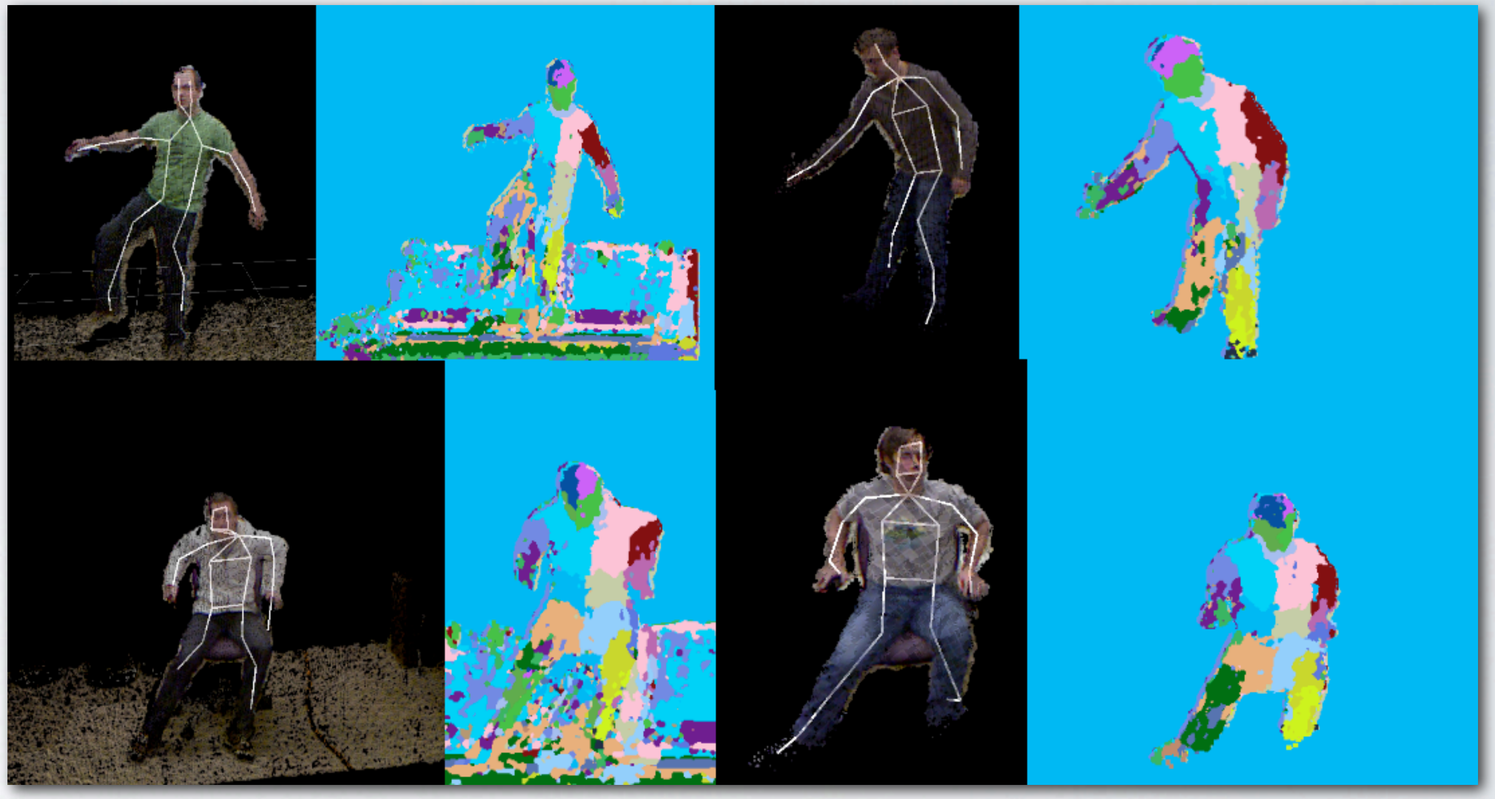
- background subtraction is difficult
- segmentation in cluttered scenes?
- need actual tracking, not just one-shot detection



People Tracking

Better solutions in PCL 1/7

- based on Random Decision Forests (RDFs)
- use additional bio-mechanical information
- fast GPU implementation



- works in 2 phases
  - offline training
  - online detection

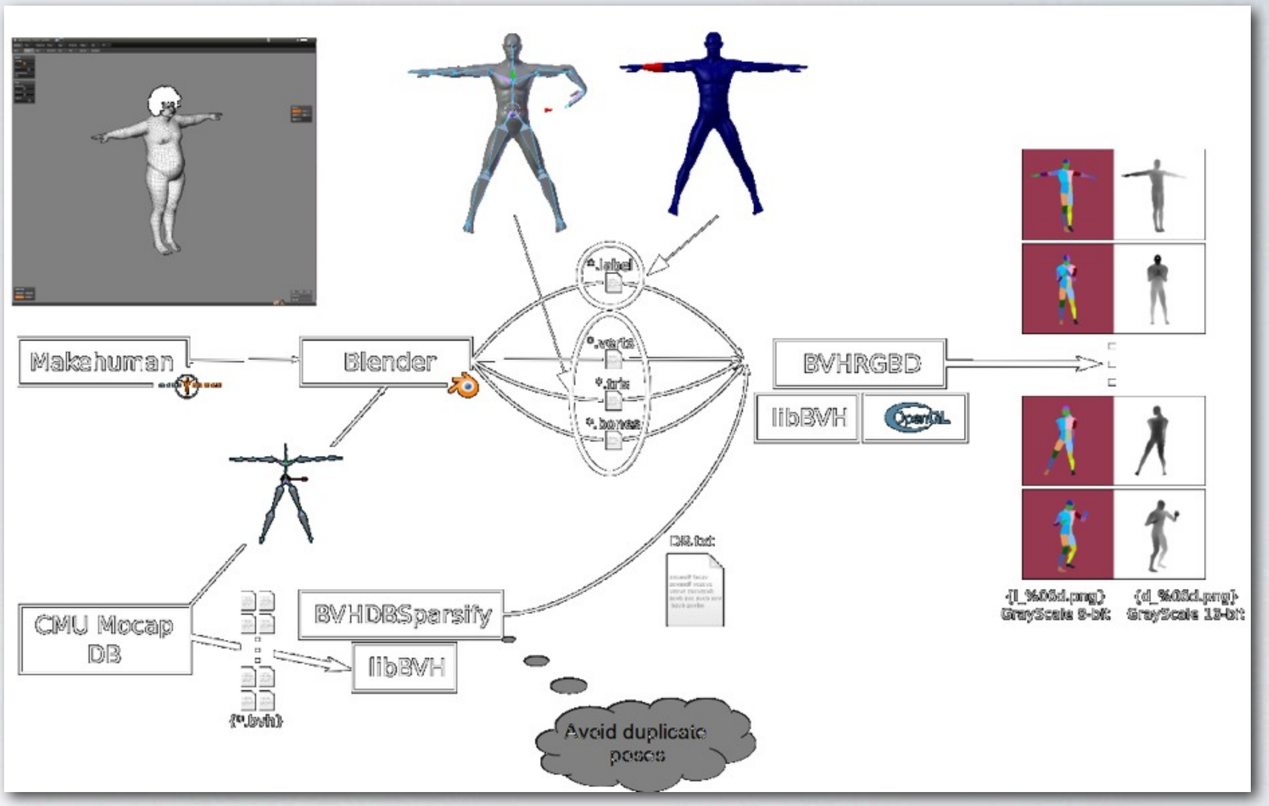
People Tracking

Better solutions in PCL 2/7

1. Offline phase:
   - build a database of human models
   - generate synthetic data using different poses and human bodies
   - train RDF trees

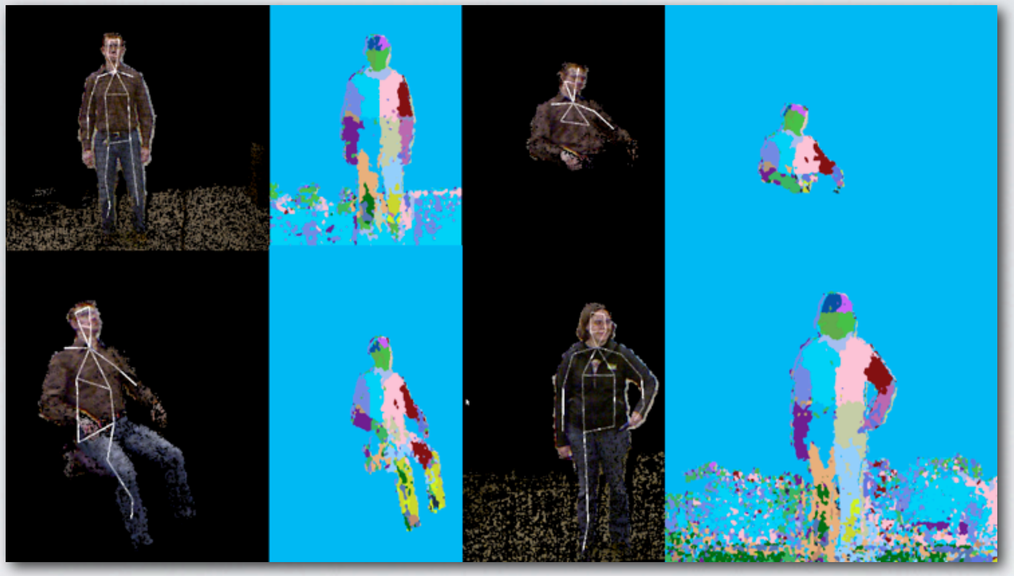
- single person with 80k poses
  - 290 GB of training data
  - 7 days of training
  - results in a training file of a few KB

use Map-Reduce tech



People Tracking

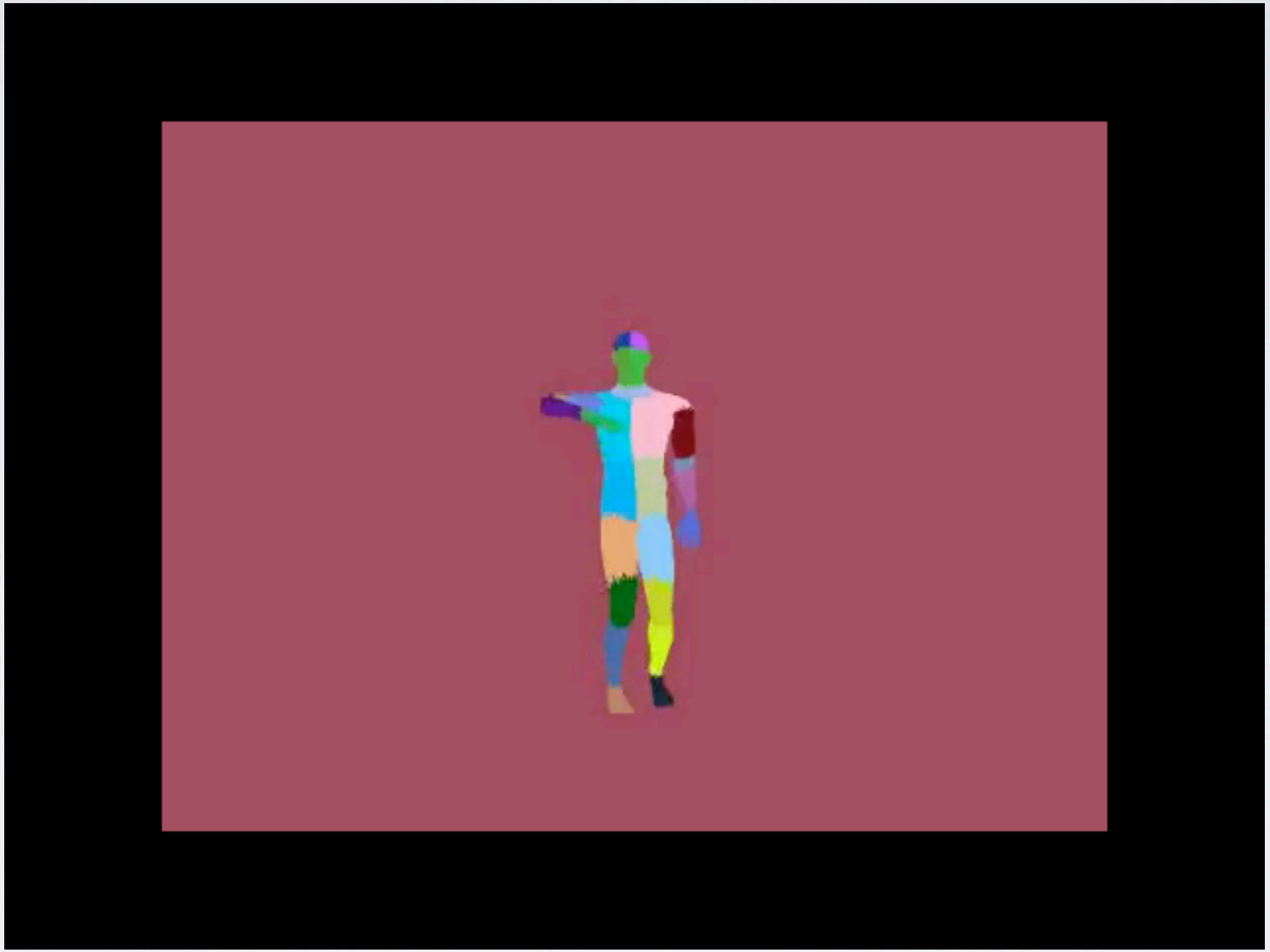
Better solutions in PCL 3/7



2. Online phase:
   - label input point clouds based on the RDFs
   - evaluate global constraints (limited limb pose space)
   - iterative refinement by imposing local consistency
   - temporal tracking

People Tracking

Better solutions in PCL 4/7

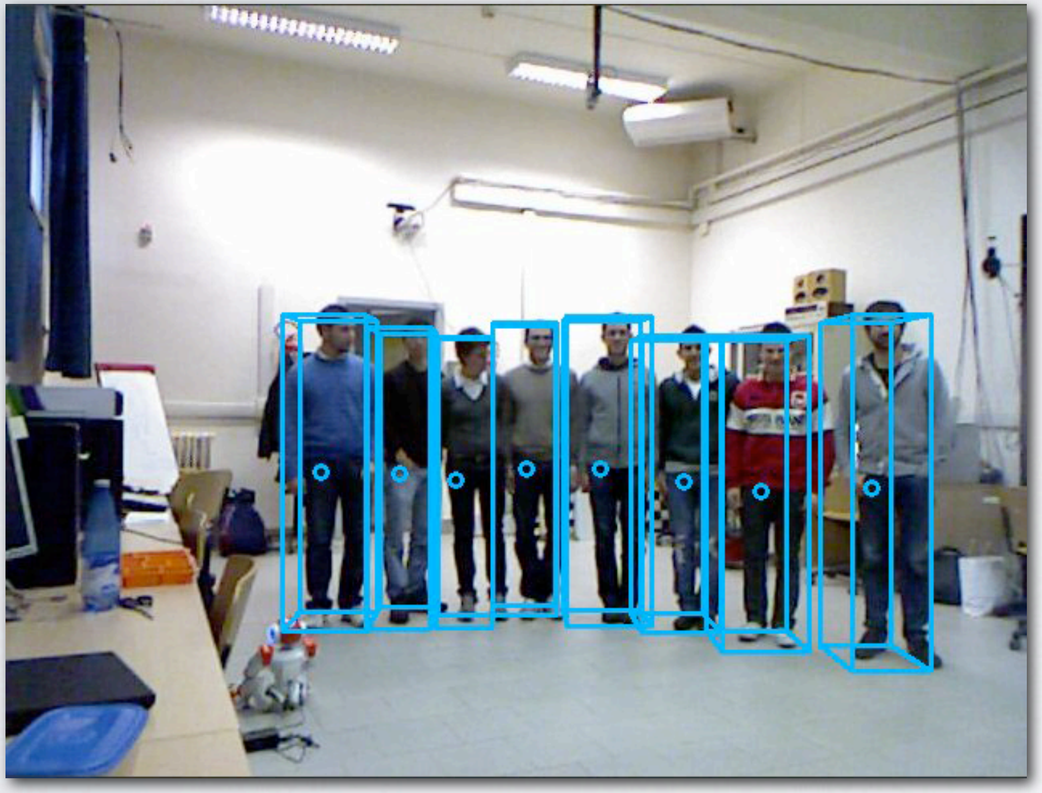


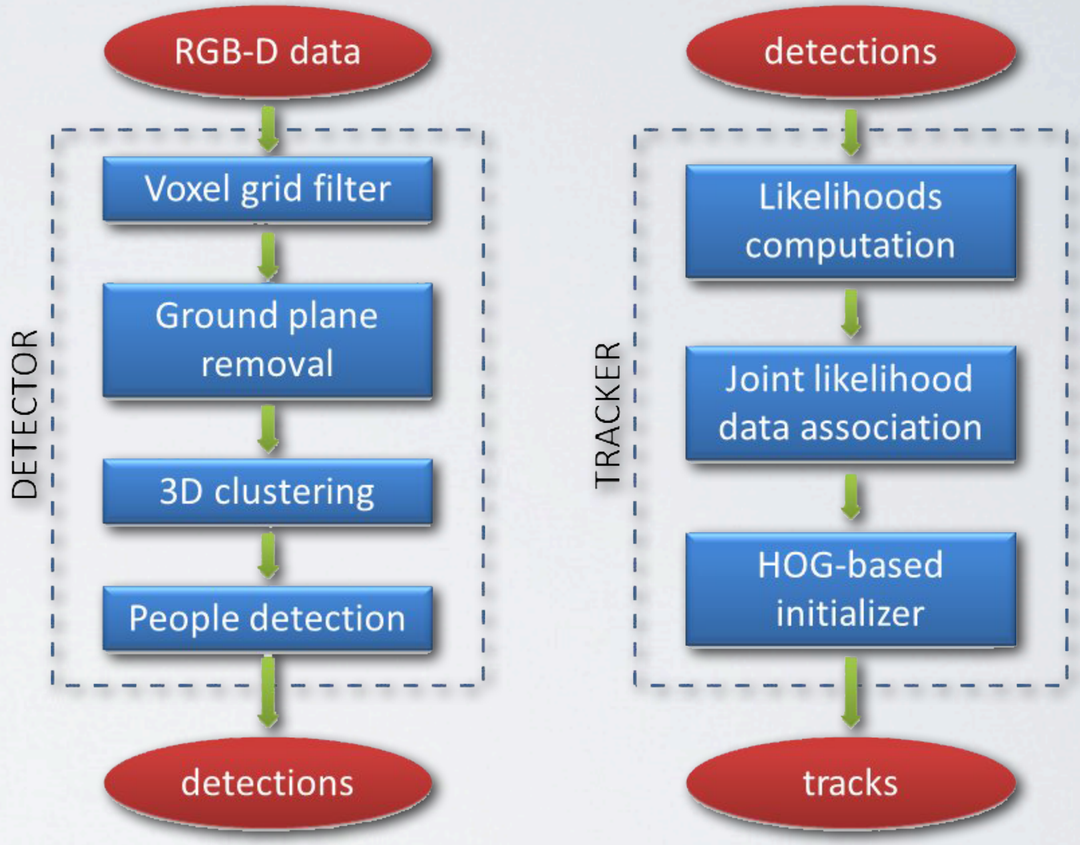
DEMO !

People Tracking

Better solutions in PCL 5/7

- People detection and tracking for groups
- CPU implementation





People Tracking

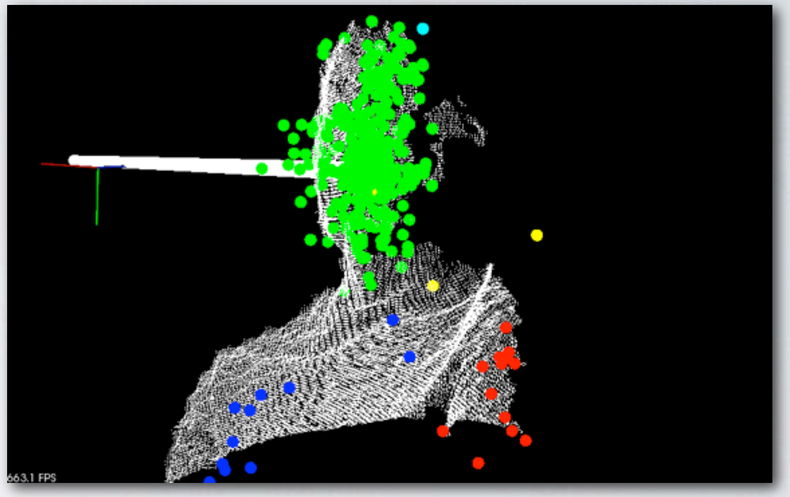
Better solutions in PCL 6/7

DEMO !

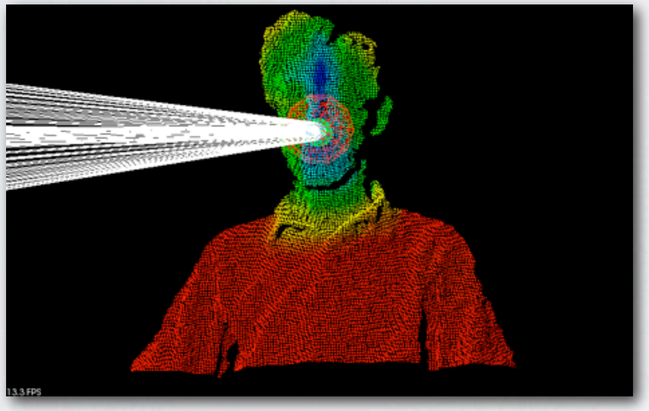
People Tracking

Better solutions in PCL 7/7

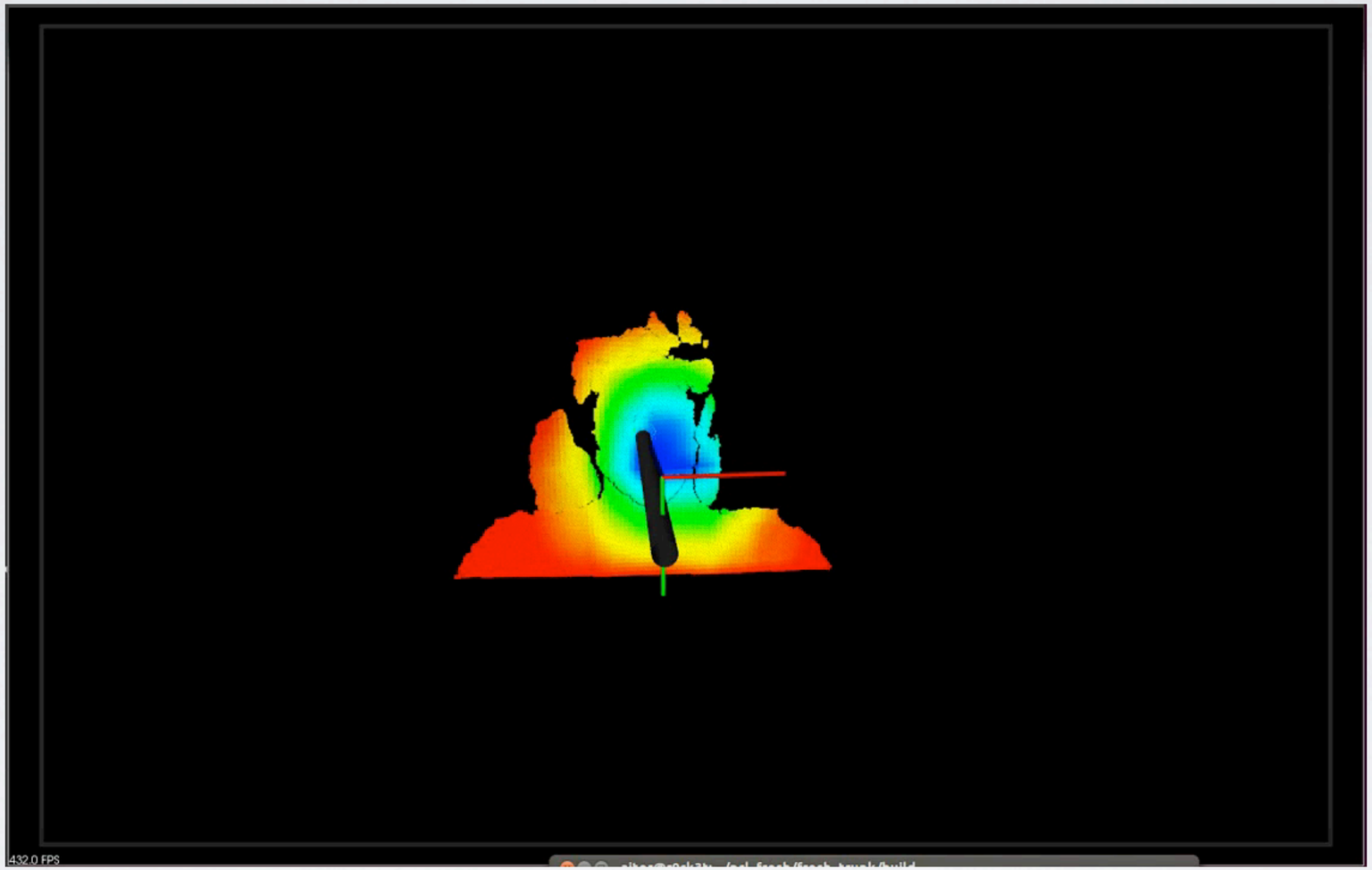
- Based on regression random forests (allow for both detection and pose estimation)
- Trained on face depth patches, head location and orientation



- Detection using a sliding window
- Cluster votes using a generic head radius
- Mean-shift cluster to remove outliers
- 6-DOF pose refinement via registration



3D Face Detection and Pose Estimation


432.0 FPS

- over 600 developers and contributors
- many thousands of users

- 100 M hits on our domains
- 500k unique

PCL Contributors


NIST
National Institute of Standards and Technology
Trimble
Velodyne
Google
Open Source
Leica
Geosystems
NVIDIA
URBAN ROBOTICS
OPEN PERCEPTION
HRI
Willow Garage
pcl
OCULAR ROBOTICS
SRI
Sandia National Laboratories
PrimeSense
Natural Interaction
DINAST
Ministry of Knowledge Economy
FOTONIC
SPECTROLAB
A BOEING COMPANY
TOYOTA

Sponsors

Job Opportunities

DONATE TO OPEN PERCEPTION FOUNDATION

Since its inception, the Point Cloud Library (PCL) project has attracted a wide range of industry attention. So much so, that many companies are now offering positions which require expertise in using and developing the PCL library as well as applications with it. Many of these not only use our software, but also wish to give back to the community. So if you love to work with PCL, and would like a great gig where you can contribute back to the project and earn money and fame doing it, please consider the following opportunities:

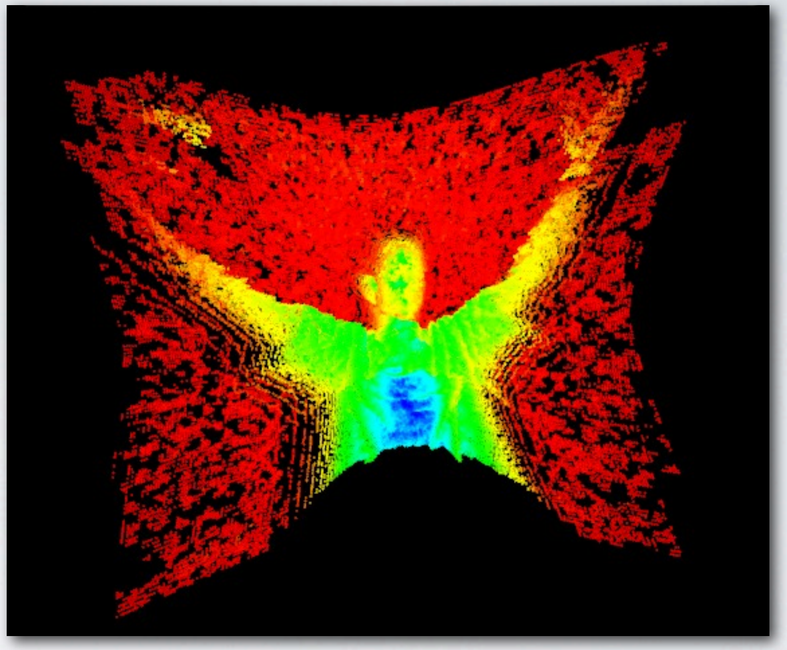
Listings

Show 10 entries Search:

Job Title	Company	Location	Posted on
Image Processing Engineer	NASA Ames Research Center	Moffett Field, CA (Silicon Valley)	3/13/2013
Chief Architect	Blue River	Sunnyvale, USA	11/30/2012
Lead Software Engineer, Point Cloud / 3D Software	Leica Geosystems	San Ramon, CA, USA	10/17/2012
Open Source Architecture & Tools Engineer	Trimble	Westminster, Colorado	10/2/2012
Computer Vision Engineer	URC Ventures	-	9/12/2012

Showing 1 to 5 of 5 entries Previous Next

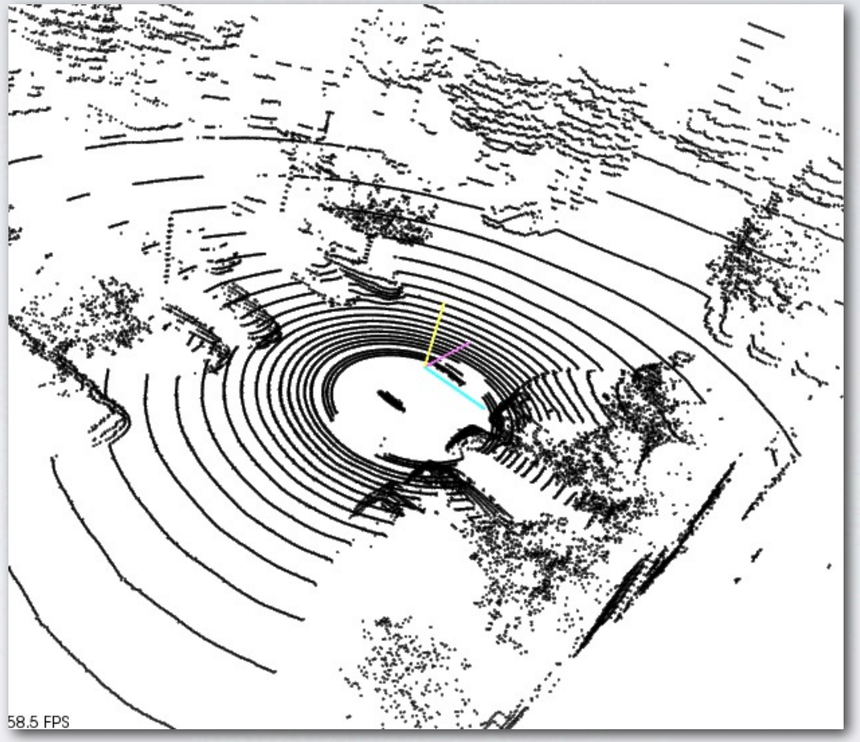
Jobs



DINAST

- Camera driver interface for DINAST cameras.
- A technology demonstrator for real-time 3D collision mapping used for motion planning with a mobile manipulator, using multiple IPA 3D cameras.

DINAST Code Sprint

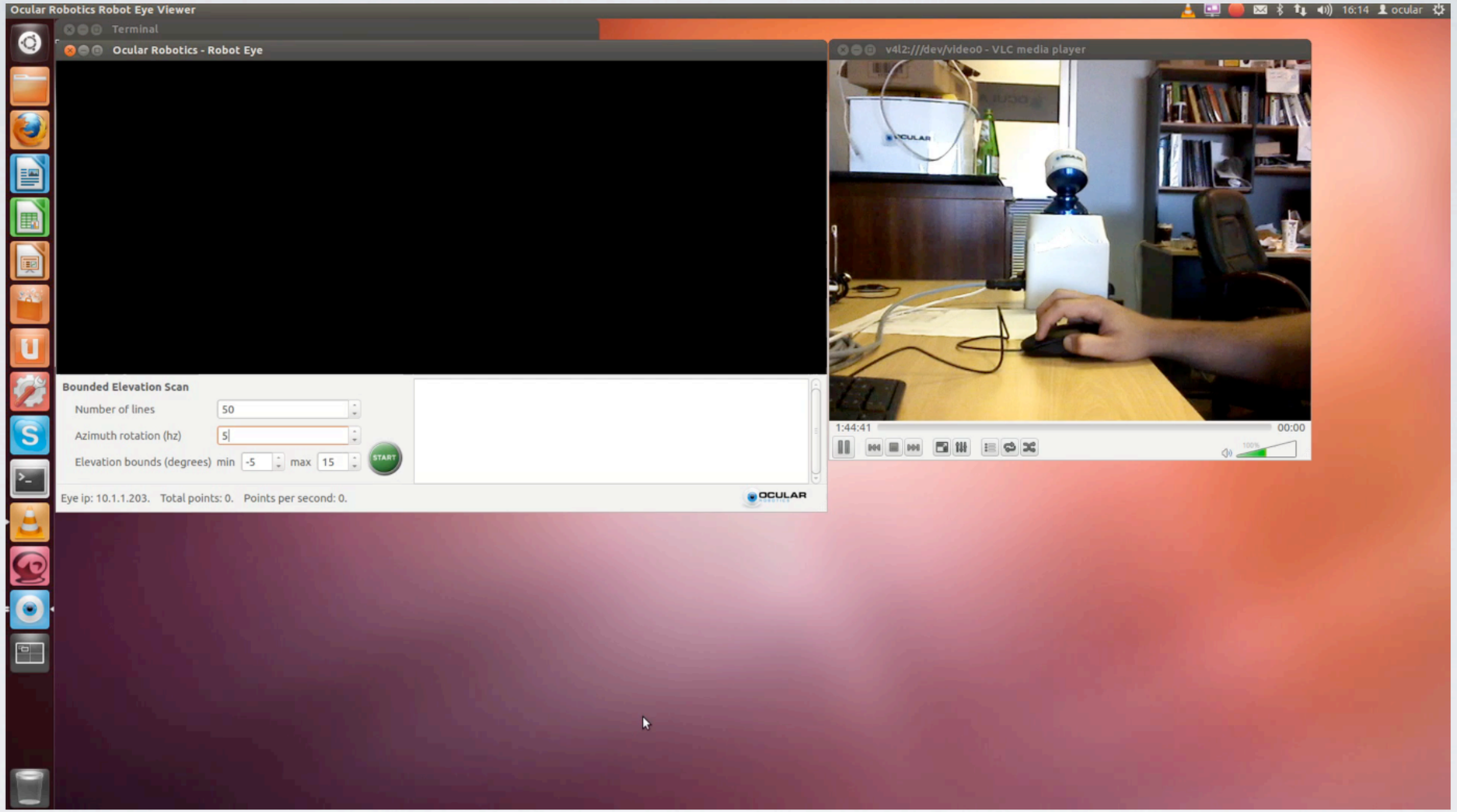


- Develop a Plug-n-Play interface for the Velodyne HDL series to make these sensors much easier to use.

Velodyne Code Sprint

- Develop an efficient PCL driver interface for the RE0x laser sensors.
- PCL Visualization module enhancements to be able to handle larger datasets and real-time visualization from high throughput sensors.

Ocular Robotics Code Sprint



Ocular Robotics Code Sprint

- Efficient compression mechanisms for organized and unorganized 3D point cloud data

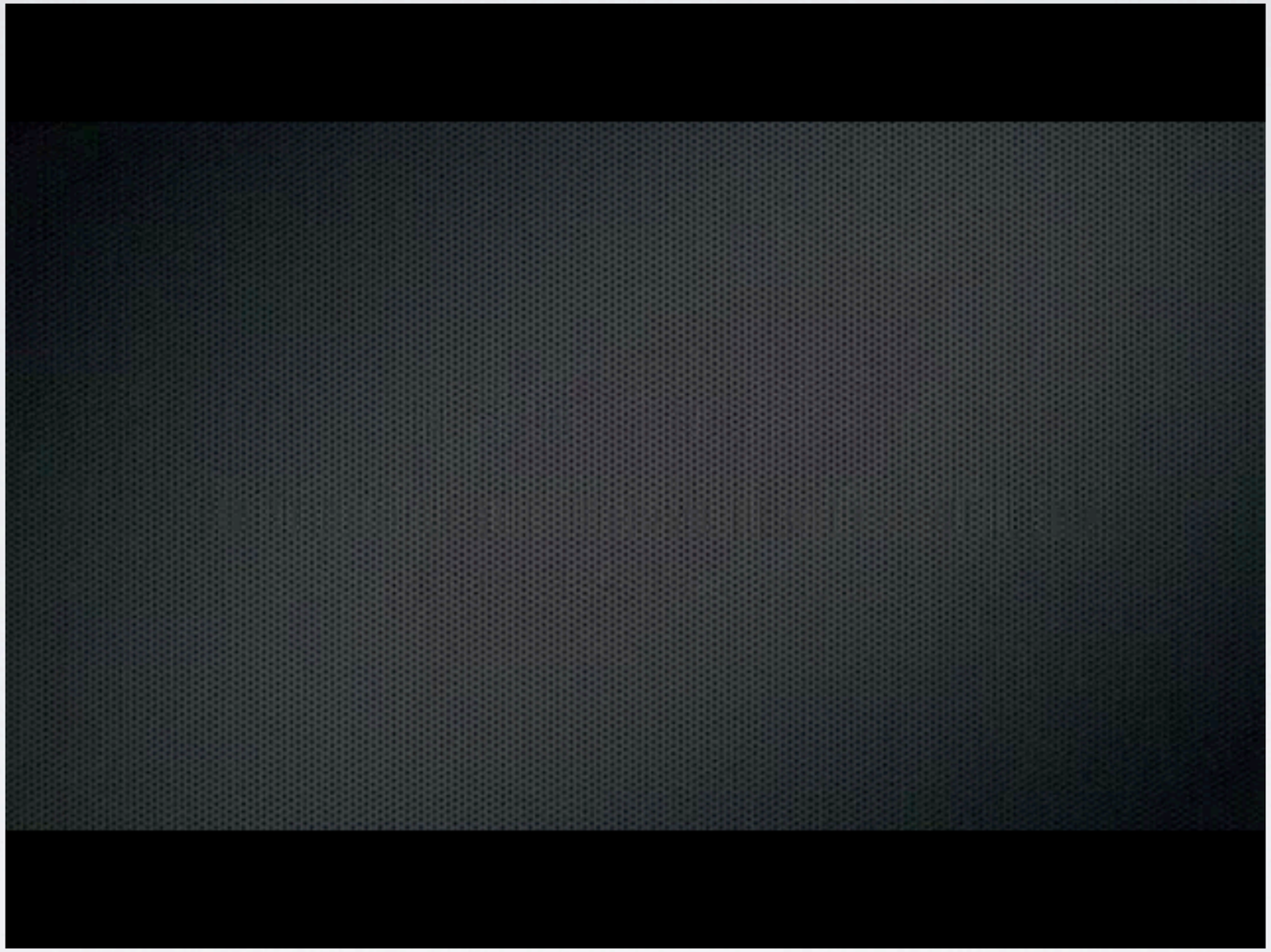
Leica Geosystems Code Sprint

2 Projects!

- Labeling outdoor pedestrian and car data as ground truth
- Fast 3D cluster recognition of pedestrians and cars in uncluttered scenes
- Part-based 3D recognition of pedestrians and cars in cluttered scenes
- Stereo-based road area detection

Honda Research Institute Code Sprint

4 Projects!



Honda Research Institute Code Sprint

4 Projects!

- Primitive shape (cylinders, spheres, cones, etc.) recognition in point cloud data
- Segmentation/Clustering of objects in cluttered environments
- 3D feature development and benchmarking

TOYOTA Code Sprints #2

3 Projects!

- Adapted visualization software and driver for Spectrolab's SpectroScan3D LIDAR imager.

Spectrolab Code Sprint

- Human detection and tracking from 2D images fused with 3D point cloud data.

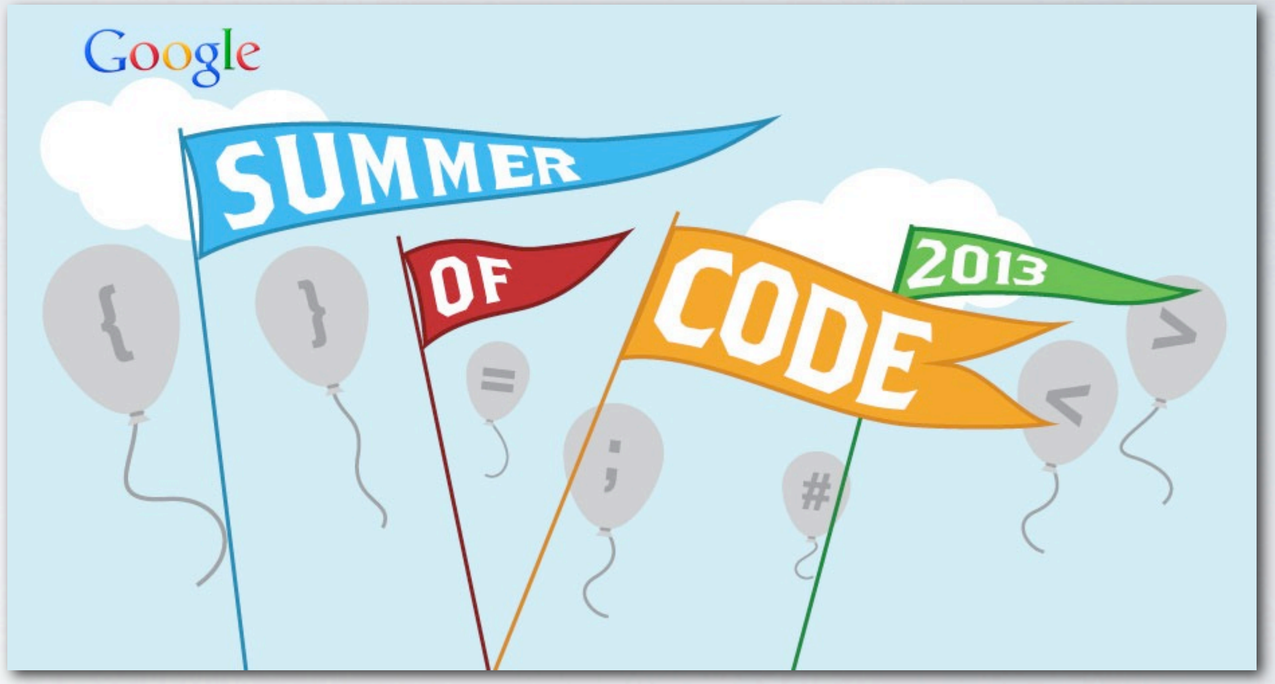
SwRI and NIST Code Sprint

- python bindings
- MATLAB The Language of Technical Computing interface for I/O and visualization
- huge dataset collection
- ParaView Parallel Visualization Application now comes with support for processing data with PCL

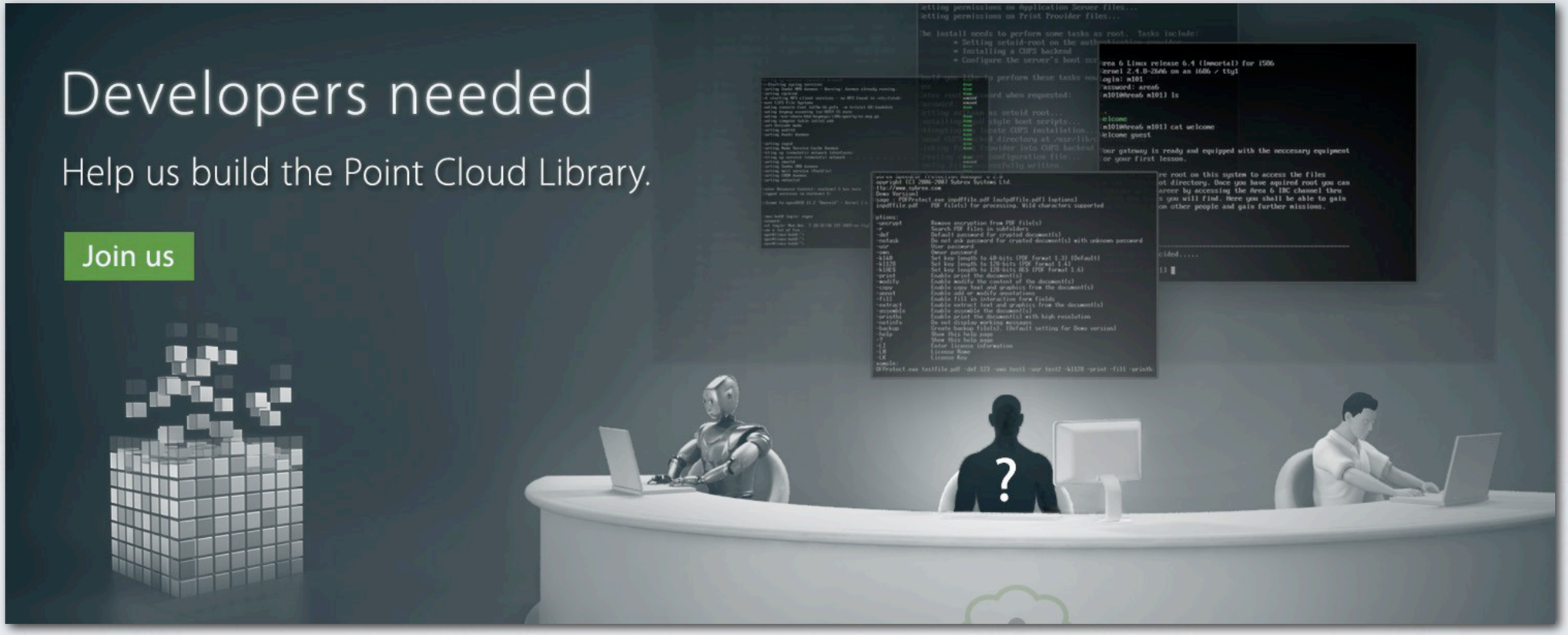
Other PCL News

- ICRA 2013 tutorial
- IROS 2013 tutorial

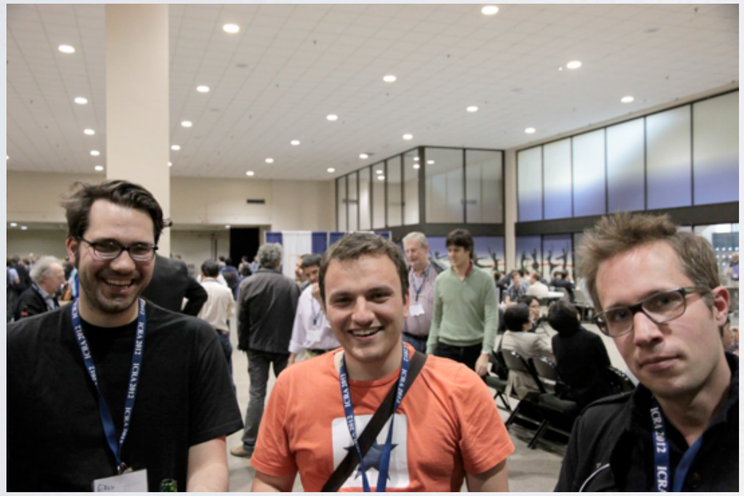
- Google Summer of Code 2013
  - 11 students last year



Upcoming Events


Developers needed
Help us build the Point Cloud Library.
Join us

pcl



Nothing could have been done without ... the awesome PCL team